Progress In Electromagnetics Research, Vol. 161, 87–100, 2018

Towards Robust Human Millimeter Wave Imaging Inspection System in Real Time with Deep Learning

Chenyu Liu[1, 2, 3], Minghui Yang[2], and Xiaowei Sun[2, *]

Abstract—With the ever-growing requirements of human security check in public, near-field millimeter wave (MMW) imaging techniques have been developing rapidly in recent years. Due to the lack of MMW images, low resolution and indistinguishable texture in most MMW images, it is still a great challenge to do high performance object detection task on MMW images. In this paper, we propose a novel framework to automatically detect concealed weapons and potential dangerous objects based on a single human millimeter wave image, in which a deep convolutional neural network (CNN) is presented to simultaneously extract features, detect suspicious objects, and give the confidence score. Unlike traditional optical image level solutions, we comprehensively analyze the original MMW data for object representation, incorporate domain-specific knowledge to design and train our network. Moreover, combined with the modern focal loss theory, we devise an effective loss function elaborately to optimize our model. Experimental results on both our dataset and real world data show the effectiveness and improvement of our method compared with the state-of-the-arts.

1. INTRODUCTION

The increasing threat of terrorism has become a safety problem as well as a serious social problem, yielding the requirements of more rigorous personnel surveillance in high-security areas, such as airports. This task usually requires high accuracy, high efficiency and low cost processing. Traditional solutions for security check consist of metal detectors, X-ray systems and a few of imaging methods. Metal detectors always need additional operators and have a long time cost, while X-ray system suffers from ionizing radiation, hence it is harmful to human body. In response, recent trends in applying non-ionizing radiation near-field imaging techniques to security check have dominated the research. These imaging methods can be further divided into two parts, active imaging sensors based and passive imaging sensors based. A series of passive imaging methods for concealed weapon detection are described thoroughly in [1], e.g., infrared imager and passive MMW imager. However, passive imagers have undesirable shortages of low resolution, faint signals, and large noise, thus they usually yield blurry images. Unlike passive imaging sensors, an active imager can achieve a high resolution, high quality, clear imaging results [2–4, 9], and is widely adopted by modern high-security facilities. In general, an MMW 3D active imager is realized by a synthetic aperture processing over a wide frequency and spatial sampling, followed by the reconstruction of the reflectivity distribution over the target spatial domain, which finally generates MMW images. Now, given a computed MMW image, how to detect concealed threats automatically with both efficiency and accuracy is the current bottleneck in the state-of-the-art MMW imaging systems.

Received 26 January 2018, Accepted 29 March 2018, Scheduled 13 April 2018

* Corresponding author: Xiaowei Sun (xwsun@mail.sim.ac.cn).
[1] University of Chinese Academy of Sciences, Beijing 100049, China. [2] Key Laboratory of Terahertz Solid Technology of Chinese Academy of Sciences, Shanghai Institute of Microsystem and Information Technology (SIMIT), Shanghai 200050, China. [3] School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China.

Traditional object detectors typically consist of two main phases, feature extraction stage and classification stage. By the successful use of handcrafted features such as SIFT [5], HOG [6], together with the statistical machine learning theory for classifiers, a big progress has been made for common vision tasks. Benefited from those previous works for optical images, a series of improvements have been made for MMW images too. [13] presents an improved EM algorithm to separate dangerous areas. [14] treats Harr-like extraction as basic features and AdaBoost as a learned classifier to search objects, and a comparison of such classical machine learning techniques for MMW images is listed in [15]. Nevertheless, these methods still cannot achieve a human level recognition results, due to the limited representation capacity of these hand-designed features.

Recently, driven by the success of deep convolutional neural networks (CNN) for image classification [18], deep learning [12] has become the most exciting approach for modern computer vision, voice and language processing tasks. The promising deep CNN is an end-to-end network, which takes an image as input and can learn deep representations from the nonlinear unions of low level features to deep features. There are two main algorithms based on CNN for object detection. One formulates object detection problem as a regression problem, in which a deep network is directly used for predicting target locations and their categories, such as YOLO [16], YOLO2 [17]; the other is a two-stage framework, which first generates region proposals and then does bounding-box regression and classification on it. RCNN [10], Fast-RCNN [11], Faster-RCNN [7] are the representative works. These methods can all obtain a human level object recognition in near real time, suggesting us to polish up the current MMW imaging systems. Yao et al. [8] firstly used a CNN in classification within a sliding window, then generated locations through a pixel-wise prediction. Though this method is useful for a small dataset, it is a kind of exhaust search, which needs a nontrivial computational cost. To push MMW image detections with deep learning efficiently, the main obstacles are: 1) compared to optical images, MMW images still suffer from low resolution and indistinguishable texture; 2) the top deep networks usually need millions of training images, but the public MMW image datasets are small and cannot be used to our specified problem; 3) the general MMW images for human are scanned without dangerous objects, which cannot be used for training a common network.

In this paper, we present a novel MMW image detection framework based on the famous two-stage Faster-RCNN pipeline. The three main contributions of our work are: Firstly, we fully explore the information reconstructed by the 3-D imaging algorithm, then analyze the feasibility of object representation, and point out an MMW image map and a spatial object map as the maximum data used for detection. Secondly, to fix the limited situation that the most available training data are images without any concealed objects, we devise our network and objective function combining with the modern focal loss theory [19] to make full use of our training data and optimize our network. Thirdly, to the best of our knowledge, it is the first work that 1) combine focal loss with a two-stage CNN detector; 2) solve this specified problem with limited training data using an end-to-end region based CNN. The following experimental results will show that our method is robust, accurate and real-time. Experiments on both our dataset and real world data show the effectiveness and improvement of our method compared with the state-of-the-arts.

2. ANALYSIS OF MMW IMAGING

2.1. Notations

In this section, symbols with an upper number 1 like $\mathbf{x}^1, \mathbf{y}^1$ denote positions at the sample transceiver, and symbols without upper numbers represent the general spatial domain. Besides, throughout this paper we will use bold symbols like $\mathbf{z}$ to denote vectors and use an upper star like p^* to denote ground-truths.

2.2. MMW Imaging Algorithm Revisited

Basic MMW imaging algorithm of our system is essentially a reconstruction process of the near-field spatial reflectivity function $f(x, y, z)$. As shown in Fig. 1, an MMW signal is firstly sent from a transceiver, then reflected from the spatial objects, and resampled in time domain at the transceiver. This procedure is carried out recursively under different frequencies, and in our system it varies from

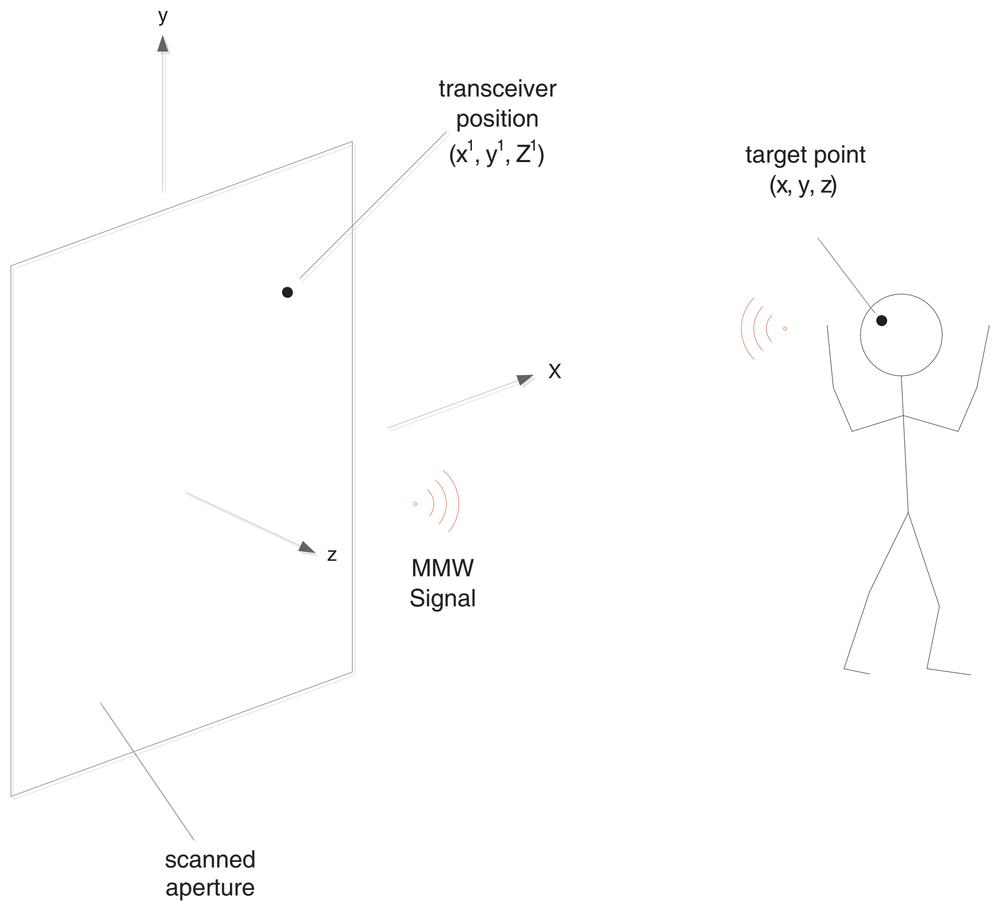

Figure 1. The 3D MMW imaging system architecture.

28 to 33 GHz linearly with total 64 frequency points sampled. At each frequency we sample 380 × 160 points at the transceiver, denoted as $s_t(x^1, y^1, t)$ in time domain and $s(x^1, y^1, \omega)$ in frequency domain. By modeling this electromagnetic physical model into mathematical, the response sampled at the transceiver in frequency domain can be formulated as:

$$s(x^1, y^1, \omega) = \iiint f(x, y, z) e^{-j2k\sqrt{(x-x^1)^2+(y-y^1)^2+(z-Z^1)^2}}\, dx\, dy\, dz, \tag{1}$$

where $k = \omega/c$ is the wavenumber, c the light speed, and ω the temporal angular frequency. After several steps of theoretical derivation, the 3D results of reflectivity distribution can be shown:

$$f(x, y, z) = \mathrm{FT}_{3\mathrm{D}}^{-1}\left\{\mathrm{FT}_{2\mathrm{D}}\left\{s(x^1, y^1, \omega)\right\} e^{-j\sqrt{4k^2-k_x^2-k_y^2}Z^1}\right\}, \tag{2}$$

where FT indicates the Fourier transformation. The detailed mathematical calculations can be found in [2], and the typical MMW image $I(x, y)$ is obtained from $I(x, y) = \max_z f(x, y, z)$. A typical MMW image for a person reconstructed from this method is shown in Fig. 2(b).

2.3. Feasibility of Object Representation

Using traditional MMW images $I(x, y)$, one can treat concealed threats detection problem as a general two-class object detection problem, a single vision task in which one class is the normal background, and the other is unsafe objects. However, it cannot exploit the full information of our reconstructed data. Basically, our imaging algorithm has two prior conditions: 1) Eq. (1) implies that the method ignores the amplitude decay, because it will have little influence on forming the image. Besides, if we consider it

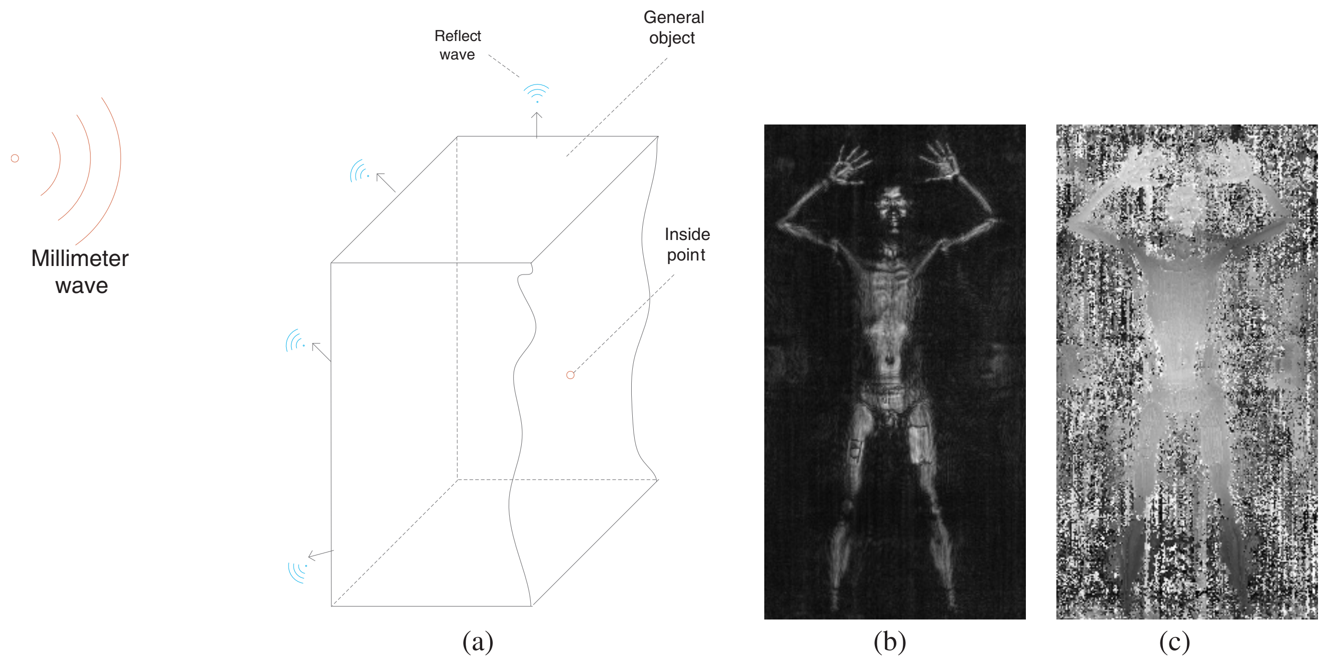

Figure 2. Feasibility of object representation: (a) A sketch map of a point inside an object. (b) An example of a general MMW image. (c) An example of our spatial object map related to (b).

thoroughly we cannot simplify Eq. (1) as Eq. (2), i.e., we are unable to solve this closed electromagnetic problem quickly using the fast Fourier algorithm; 2) The reconstructed information cannot picture the full ground-truth 3D data of object, as shown in Fig. 2(a): when a point is considered inside a dense closed object, MMW cannot pass to this point, indicating that the signal received at the transceiver is only the sum of all signals reflected from the most achieving surface points. In other words, the method widely used before is a kind of reflectivity surface reconstruction algorithm. We denote $f_{tr}(x, y, z)$ as the actual 3D reflectivity function, then our reconstructed $f(x, y, z)$ should be:

$$f(x, y, z) = S(x, y, z) \cdot f_{tr}(x, y, z), \tag{3}$$

where $S(x, y, z) = 1$ if MMW can spread to point (x, y, z), and $S(x, y, z) = 0$ if it cannot reach it. It suggests that our reconstructed $f(x, y, z)$ can only sketch the outer borderline of objects, rather than imply the full 3-D field's reflectivity.

Unlike most previous works, here we present an extra spatial object map $D(x, y)$ combined with the visual image $I(x, y)$ to represent objects. Actually, MMW image is lack of textures, colors as well as resolutions compared to optical images. Nevertheless, as we analyze in this subsection, our imaging result still carries the spatial information. The surface information can be viewed as 2.5-D data, because for a viewing pixel in $I(x, y)$, the extra as well as the only message we could extract is the depth information. The depth for one pixel indicates the distance between the focusing plane of the holographic reconstructed points and our transceiver, modeled as $D(x, y) = \arg\max_z f(x, y, z)$. Fig. 2(c) is a visualization example of spatial depth object map. Based on $I(x, y)$ and $D(x, y)$, it is able to extract all useful information using merely two simple maps, while both simplify our imaging results and preserve feasibility of object representations.

3. OUR DETECTION APPROACH

3.1. Overview of Methodology

The proposed method for our specific task is improved based on the deep learning framework Faster RCNN [7], which is proved as a state-of-the-art scheme for object detection. This pipeline consists of two stages: 1) a Region Proposal Network (RPN) or called Regions of Interests (RoIs), which generates a series of regions that are likely to contain objects; and 2) a Fast RCNN [11] network for further object classification and region boundary refining. In this paper, we extend this architecture to our threats detection problem, and the main difference is that we are trying to learn the normal background representations in order to detect potential threats, instead of classifying regions into specific objects.

A general classifier is usually trained from supervised learning, which focuses on teaching the label information to the classifier and minimizes the designed loss function; therefore, it has learned specific feature mapping for each object. When it comes to our problem, by considering how an operator looks, thinks and recognizes, one simple solution is to learn the normal views of MMW data like humans do, then our model can localize unsafe regions naturally. Motivated by this, we propose to train our model end to end by the following procedure, while making full use of our relatively more unlabeled background data, and the whole framework is shown in Fig. 3.

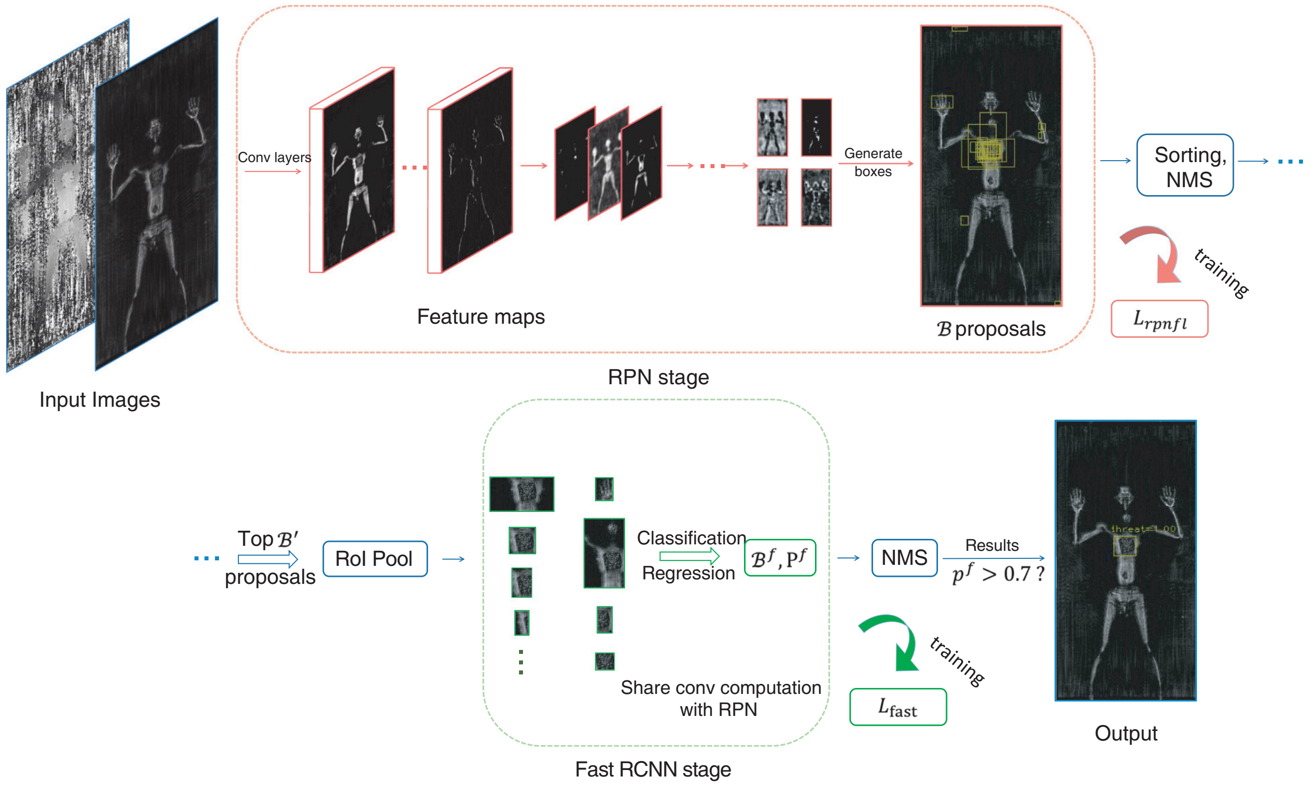

Figure 3. Whole framework of our detection approach. Our deep network firstly takes the two-channel image as an input, and the next RPN stage will generate $\mathcal{B}$ potential region proposals. After sorting and a NMS, the rest $\mathcal{B}'$ regions with its extracted RoI features is transported into the later Fast RCNN stage. Finally, the overlapped RCNN refined results is reduced by the other NMS, then our model outputs the predicted regions whose probability is higher than 0.7. Our model is trained end-to-end by the designed multi-stage loss L_{rpnfl} and L_{fast}.

When training, first of all, our experimental CNN models are all pretrained from the optical image dataset ImageNet [20], then we begin the further finetune stage based on Faster-RCNN and our two-channels input $I(x, y)$ and $D(x, y)$. During training in RPN stage, the original loss is designed for labeled training images with randomly generated balanced anchor boxes [7]. However, because our MMW images are collected and calibrated in public places, nearly half of our MMW data are taken without any threats, thus: 1) those images cannot be used for training under the original framework; 2) we cannot learn the inherent features of background infos. To overcome it, we devise a new RPN loss function L_{rpnfl} for our task, which combine with the most exciting focal loss [19] to optimize RPNs.

After RPN, the network will produce many region proposals $\mathcal{B}$ with its probability scores $\mathcal{P}$ of being an concealed object, and a non maximum suppression (NMS) step is applied to remove low score and overlapped boxes. At the latter Fast RCNN stage, the original network and cross entropy loss are preserved to classify and refine the RPN results $\mathcal{B}'$, which predicts the final boxes $\mathcal{B}^f$ and scores $\mathcal{P}^f$. Lastly, our model applies a total loss L_{fast} of Fast RCNN in [11] to optimize the whole network. Under this improvement, one can directly train Faster-RCNN with unlabeled normal images, yielding a great improvement of the original methods.

In practical testing, our network takes the two-channel image into RPN network and will generate thousands of region proposals with their confidence scores. Our model will keep the top 2000 boxes according to their scores, then an NMS is processed to remove the overlapped proposals. Then the kept n_{fast} (60 in our experiments) image patches are imported into the latter Fast RCNN in parallel, in order to generate the refined boxes. Furthermore, our algorithm employs the other NMS to those refined n_{fast} boxes, and the final results are those boxes whose confidence scores are greater than the fixed threshold p^* (we set it to 0.7 in our experiments).

3.2. Models and Loss Functions

In the following experiments, we apply ZF (5 conventional layers, 3 fully connected layers) [22] and VGG16 (13 conventional layers, 3 fully connected layers) [23] as our base network respectively to prove the effectiveness of our detection approach. The detailed settings of our networks and training sketches will be discussed in the next section. In this subsection, we mainly focus on how we design our models and how it works.

3.2.1. Region Proposal Stage

An RPN takes our two-channel image as input and outputs a series of rectangular region proposals, each with a probability score for being a concealed object. Following [7], for each image in our training set, we obtain totally WHk anchors for generating positive and negative examples. On the last convolutional feature map with size $W \times H$, each pixel is assigned k anchors with different scales and aspects. For each anchor, it will be assigned as a positive example when it has an Intersection-over-Union (IoU) overlap with any ground-truth box higher than 0.6, and be treated as a negative example if its IoU is lower than 0.3.

During training, after all anchors are assigned with labels in RPN stage, we minimize a multi-task loss function that consists of classification loss and regression loss. The general objective function for RPN of an anchor box B is defined as:

$$L_{rpn}(p, p^*, \mathbf{t}, \mathbf{t}^*) = L_{reg}(p^*, \mathbf{t}, \mathbf{t}^*) + L_{cls}(p, p^*), \tag{4}$$

where p denotes the predicted probability of B, and $\mathbf{t} = (t_x, t_y, t_w, t_h)^T$ is a predicted four-dimensional vector standing for the coordinates transformation parameters for bounding box regression, see Appendix A.2. The RPN ground-truth label p^* is 1 if B is assigned positive and $p^* = 0$ when B is a negative example, and $\mathbf{t}^*$ is the ground-truth parameters for a positive anchor.

When it comes to a positive anchor box B, the first term L_{reg} in Eq. (4) tries to minimize the distance between our predicted location and ground-truth location. In contrast, if it is a negative example, we just need the predicted probability p to be small or equal to 0, thus we ignore this item naturally. Hence L_{reg} can be written as:

$$L_{reg}(p^*, \mathbf{t}, \mathbf{t}^*) = p^* \mathrm{F}_{\mathrm{SML1}}(\mathbf{t} - \mathbf{t}^*), \tag{5}$$

where the distance function for vectors $\mathrm{F}_{\mathrm{SML1}}(\cdot)$ is the robust smooth L_1 function, and the detailed definition and derivative of $\mathrm{F}_{\mathrm{SML1}}(\cdot)$ can be found in Appendix A.1.

The second term L_{cls} is designed for minimizing the classification error, if $p^* = 1$ our predicted p should be close to 1, and vice versa. Most classification tasks use a standard cross entropy log loss for training, denoted as:

$$L_{cls}(p, p^*) = p^* \log p + (1 - p^*) \log (1 - p). \tag{6}$$

However, in MMW detection tasks, our training set is unbalanced because: 1) many images do not have any threats, thus our two-stage networks cannot generate any positive examples; 2) even if an input has some positive ground-truth regions, it is still quite unbalanced because the concealed objects usually take just a small region, leading to the unbalanced numbers of assigned anchor boxes (e.g., 2 positive vs. 1022 negative examples). Optimization with L_{cls} in our problem will converge to a bad local minima since unbalanced negative samples will dominate the gradients while training. This obstacle also highly affects one-stage detectors like YOLO, for its extreme dense assigned anchors for direct regression. Surprisingly, [19] proposes to solve the class imbalance by down-weighting the loss assigned to well and easy classified examples and adopt a dynamically scaled cross entropy loss called focal loss where the scaling factor decays to zero as confidence in the correct class increases. Mathematically, the new loss is:

$$L_{cls\_fl}(p,p^*) = \alpha p^*(1-p)^\gamma \log p + (1-\alpha)(1-p^*)p^\gamma \log(1-p), \tag{7}$$

where $\alpha \in [0,1]$ is a weighting factor for unbalanced class, and the focusing hyper-parameter γ smoothly changes the rate at which easy examples are down-weighted. Under the setting of $L_{cls\_fl}(p,p^*)$, we now extend the one-stage focal loss to our unbalanced and unlabeled training sets. While training RPN, we replace $L_{cls}(p,p^*)$ by $L_{cls\_fl}(p,p^*)$. Therefore, if an input has an object, we can easily generate 1:100 positive vs. negative examples for learning, and if it does not have any threats, we randomly sample a series of anchors as the desired easy classified negative training examples instead. This training framework follows the initial assumption that our model should learn like how human learns, for the purpose of learning from normal or safe distributions to automatically detect unsafe objects. So the overall objective function of our RPN network is:

$$\begin{aligned} L_{rpnfl}(p,p^*,\mathbf{t},\mathbf{t}^*) &= \lambda L_{reg}(p^*,\mathbf{t},\mathbf{t}^*) + L_{cls\_fl}(p,p^*) \\ &= \lambda p^* \mathrm{F}_{\mathrm{SML1}}(\mathbf{t}-\mathbf{t}^*) + \alpha p^*(1-p)^\gamma \log p + (1-\alpha)(1-p^*)p^\gamma \log(1-p). \end{aligned} \tag{8}$$

The hyper-parameter λ in Eq. (8) is a weighted factor for balancing training numbers N_{cls} and N_{reg}, where $N_{reg} \ll N_{cls}$ because $L_{reg}(p^*,\mathbf{t},\mathbf{t}^*)$ will be 0 if the train box is negative (i.e., $p^*=0$). For optimizing our network, we use back propagation and the first order method stochastic gradient descent (SGD) [21]. The gradient w.r.t. to our loss function input is computed as:

$$\begin{aligned} &\frac{\partial L_{rpnfl}}{\partial t_i} = \lambda p^* \mathrm{G}_{\mathrm{SML1}}(t_i - t_i^*), \quad t_i \in (t_x, t_y, t_w, t_h)^T, \\ &\frac{\partial L_{rpnfl}}{\partial p} = \alpha p^* \left[-\gamma(1-p)^{\gamma-1}\log p + \frac{(1-p)^\gamma}{p}\right] + (1-\alpha)(1-p^*)\left[\gamma p^{\gamma-1}\log(1-p) - \frac{p^\gamma}{1-p}\right], \end{aligned} \tag{9}$$

where $\mathrm{G}_{\mathrm{SML1}}$ denotes the derivative of point-wise $\mathrm{F}_{\mathrm{SML1}}$, see Appendix A.1.

3.2.2. Fast RCNN Stage

After RPN, there are many predicted boxes $\mathcal{B}$ which are likely to contain objects, each with a confidence score $\mathcal{S}_j$. The number of these potential boxes is huge (e.g., near 19000 in ZF [22] net), and many easy negative boxes and overlapped boxes are removed by the latter score sorting and NMS. Then we randomly choose 128 patches from the rest boxes $\mathcal{B}'$ as input to next RCNN network through RoI pooling [7], for refining and classifying those patches, and compute final loss function to jointly train our network (see the second stage in Fig. 3). We sample 128 boxes from $\mathcal{B}'$ based on the principle that numbers of positive and negative examples should come to $1:1$, if $\mathcal{B}'$ does not have 64 positive boxes, we pad it with negatives. This stage is extremely fast both in training and testing, since the RCNN network shares most computation with the previous RPN stage, so the overall computation cost is nearly equal to a general CNN. In this stage, an input patch from $\mathcal{B}'$ will be treated as threat region if the IoU is higher than 0.5, and is treated as normal region when the IoU is lower than 0.3.

For a final output region $\{p_f, \mathbf{t}_f, p_f^*, \mathbf{t}_f^*\}$, the loss function of Fast RCNN is similar to L_{rpnfl} except the densely focal loss:

$$\begin{aligned} L_{fast}(p_f,p_f^*,\mathbf{t}_f,\mathbf{t}_f^*) &= L_{reg}(p_f^*,\mathbf{t}_f,\mathbf{t}_f^*) + L_{cls}(p_f,p_f^*) \\ &= p_f^* \mathrm{F}_{\mathrm{SML1}}(\mathbf{t}_f - \mathbf{t}_f^*) + p_f^* \log p_f + (1-p_f^*)\log(1-p_f). \end{aligned} \tag{10}$$

The gradient w.r.t. to $\{p_f, \mathbf{t}_f, p_f^*, \mathbf{t}_f^*\}$ can be deduced by set $\lambda = 1$, $\alpha = 0.5$ and $\gamma = 0$ in Eq. (11):

$$\frac{\partial L_{fast}}{\partial t_{fi}} = p_f^* \mathrm{G}_{\mathrm{SML1}}(t_{fi} - t_{fi}^*), \qquad \frac{\partial L_{fast}}{\partial p_f} = \frac{\alpha p_f^*}{p_f} + (1-\alpha)\frac{1-p_f^*}{1-p_f}. \tag{11}$$

Combining the objective function of RPN and Fast RCNN stages, the overall loss of our network L_{all} can be written as Eq. (12) based on Eq. (8) and Eq. (10):

$$L_{all} = L_{rpnfl} + L_{fast}. \tag{12}$$

By the definition of our models and loss functions, we can train our whole network end-to-end by mini-batch SGD now.

4. EXPERIMENTS

4.1. Datasets and Evaluation Protocol

We comprehensively validate the effectiveness of our proposed method on two datasets: 1) a standard MMW image dataset designed by SIMIT [9], called SIM dataset; and 2) a small set of MMW images collected at real world (RW) security checkpoints, named RW dataset. Note that all images described here represent the entire MMW imaging data, which contain $I(x, y)$ and $D(x, y)$ discussed in Section 2.3. The SIM dataset contains totally 6,030 two-channel images, which are divided into 2 categories: 3,003 images are labeled manually with their threat region positions, while the rest are all normal examples. SIM dataset was collected strictly for data analysis: we invited hundreds of volunteers wearing different clothing types, investigated different potential dangerous objects, and placed it on almost all positions on human body to build this big dataset. The RW dataset consists of 400 MMW images collected from security checkpoints, and each is associated with labels and regions by an operator. RW dataset is designed for evaluating different algorithms practically. All images in SIM and RW dataset are collected through our imaging systems [9], with a fixed pixel size of 380×190. Visualized examples of SIM dataset and RW dataset can be found in Fig. 4 and Fig. 5.

We compare our method with several state-of-the-art MMW and optical image detection baselines, which can be divided into two parts: classical machine learning based methods and end-to-end deep learning based methods. For classical machine learning based methods, we investigate two representative models in MMW images, including AdaBoost [14] and exhausted CNN [8], each implemented with a sliding window. For end-to-end deep learning frameworks, we apply the exciting one-stage detection network YOLO2 [17] and the original Faster RCNN [7] under different depths of networks: ZF [22], VGG16 [23], Res-50 [24].

Following the measurements in daily security check and the protocol in [8], we employ precision (PRCS) and recall (RC) to evaluate the performance of the proposed method and other baselines. Precision is designed for judging the correctness ratio of a detector, and recall is used for judging the detection ratio of a detector. Precision and recall are mathematically writing as:

$$\text{PRCS} = \frac{\#\{\text{right predicted}\}}{\#\{\text{total predicted}\}}, \qquad \text{RC} = \frac{\#\{\text{right detected}\}}{\#\{\text{total threats}\}}. \tag{13}$$

4.2. Experimental Settings

We randomly select 5,030 images in SIM dataset as training set, and naturally, the rest 1,000 images from SIM dataset and the whole 400 images from RW dataset constitute our testing set, totally 1,400 images.

For machine learning based methods, there is a window of size 28×28 slide from the original 380×190 image, which extracts features and does classification task for each pixel location. In AdaBoost model, Harr-like features are applied to extract information from each small window, and we use a trained AdaBoost classifier to vote each pixel into objects, while in exhausted CNN, a CNN which takes a 28×28 image as input is designed for classifying each pixel location.

For end-to-end deep learning methods, YOLO2 firstly resizes the input images to 480×480, then we train the network following the same configuration in its original experiment [16]. In Faster RCNN, we firstly resize the images to 1000×500, then train it under its original loss function L_{rpn} and L_{fast} [7]. We evaluate three deep models with Faster RCNN: 1) 8-layer network ZF-net; 2) 16-layer network VGG16-net; 3) 50-layer network Res50-net. In addition, we implement these described methods under our experimental settings based on the codes provided by the authors.

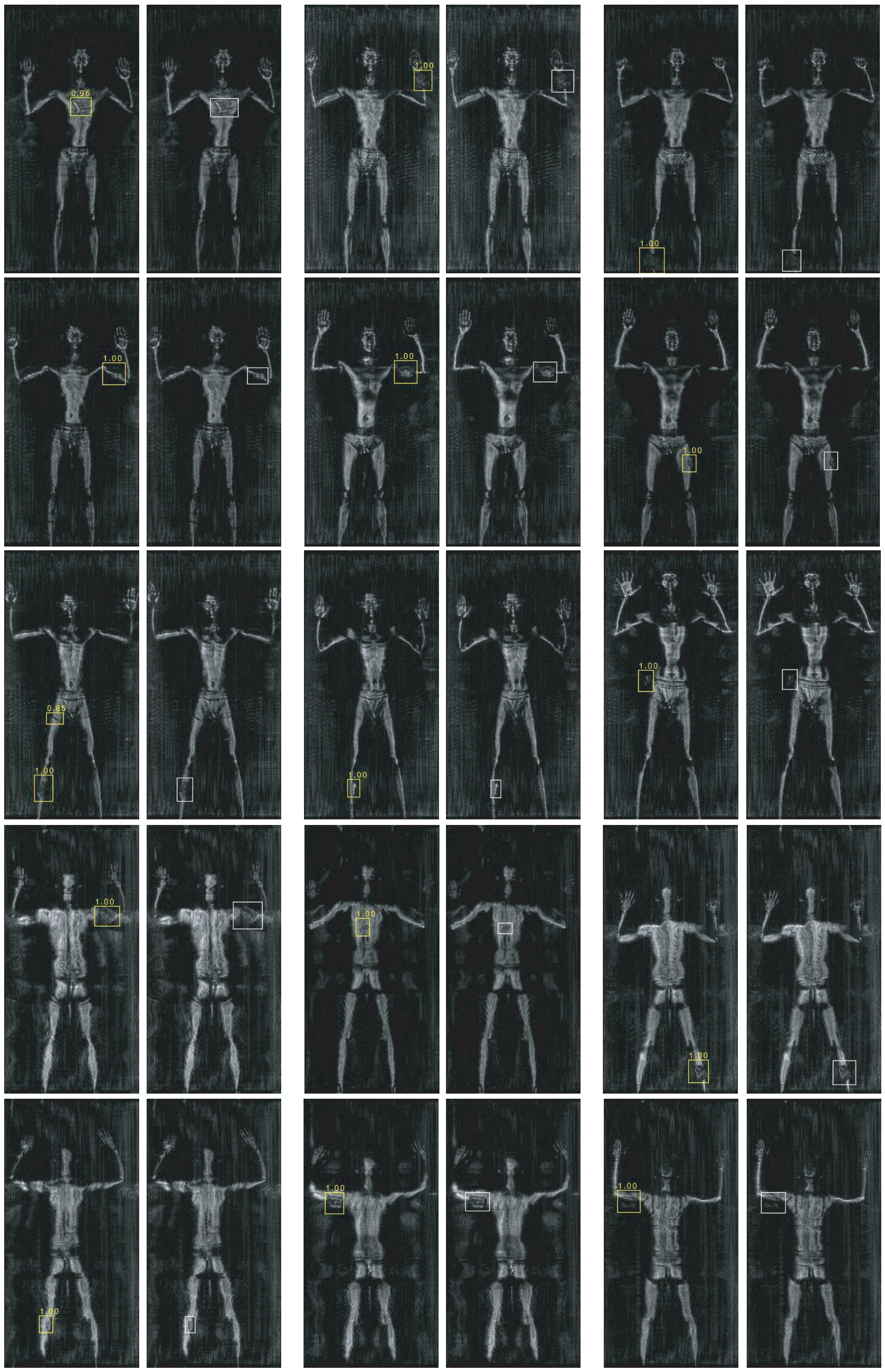

Figure 4. Selected testing examples of concealed threat detection results of SIM dataset using our proposed framework. We use a slight ZF model in Table 1 Proposed-ZF to obtain these results. Each predicted box is assigned with a confidence score in $[0, 1]$, and we apply a score threshold of 0.85 to display these images. The first three lines are front view of human examples, and the latter lines are back view. For a pair of images in this figure, the left image with yellow boxes and scores is our predicted results, and the right image with white boxes is the original image with manual labeled ground-truth.

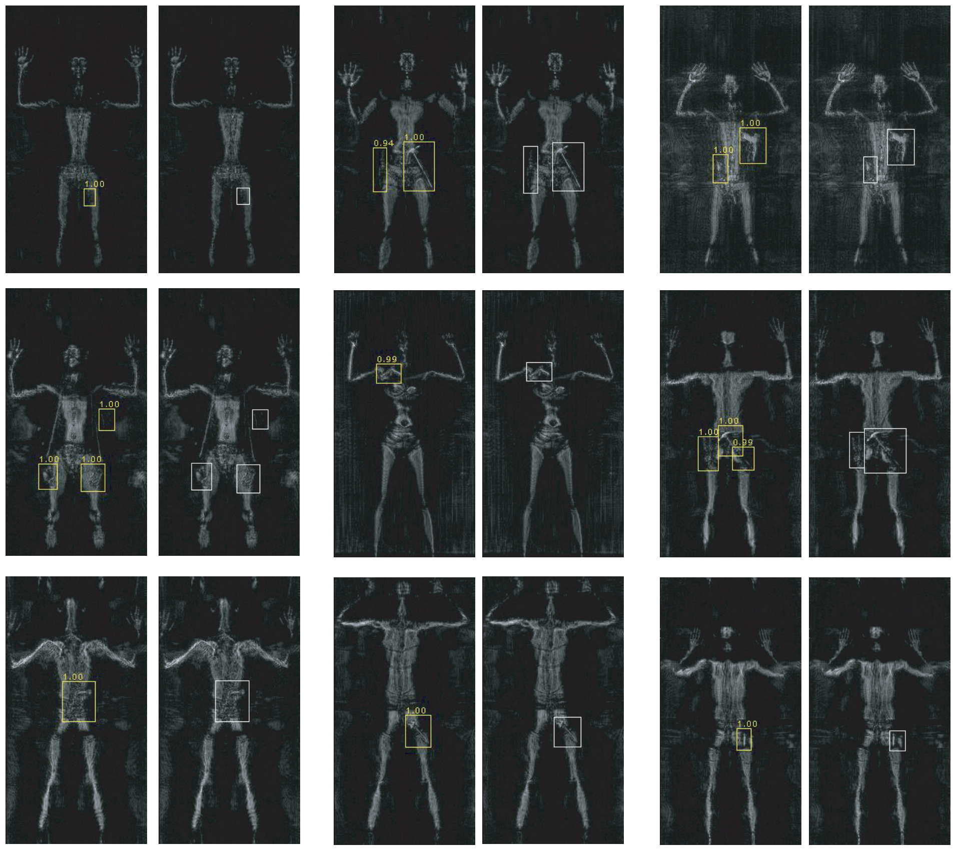

Figure 5. Selected testing examples of concealed threat detection results of RW dataset using our proposed framework. We use a slight ZF model in Table 1 Proposed-ZF to obtain these results. Each predicted box is assigned with a confidence score in $[0, 1]$, and we apply a score threshold of 0.85 to display these images. For a pair of images in this figure, the left image with yellow line boxes and scores is our predicted results, and the right image with white boxes is the original image with manual labeled ground-truth.

In our method, benefited from the proposed L_{rpnfl}, we can train RPN with the unbalanced and dense training proposals, so we randomly choose 4800 anchor boxes for training RPN (vs. originally 256 anchors for training). The coefficient α in Eq. (8) is heuristically set to 0.25, and the weighting factor γ is set to 2 during training. All experiments for our model are implemented with Caffe [25] on a desktop PC with a NVIDIA GTX1070 GPU. For SGD we set the weight decay to 10^{-5}, momentum to 0.9, and we use a mini-batch size of 128 in Fast RCNN stage. We start with a learning rate of 0.001, multiply it by 0.1 at 50,000 iterations, and terminate training at 75,000 iterations.

4.3. Performance Evaluation

4.3.1. Main Results

Table 1 shows results of our method compared with other baselines for MMW image processing and detection. Using RPNFL+VGG, we can obtain 94.85% PRCS and 90.95% RC within a running time of 108 ms, achieving the best performance among all previous works. For RPNFL+ZF, the results are 93.85% for PRCS and 90.15% for RC, slightly lower than the 16-layers network VGG, but it has a extremely fast detection time of 29 ms. From the direct view of Table 1, we can observe that: 1) although MMW images do not have rich information apparently, deep learning based methods still outperform other common techniques within a large gap; 2) for our anomaly detection problem, two-

Table 1. Comparison of detection results on the test set. Our proposed method reaches a new exciting accuracy with a shot time of 60/120 ms. Notice that the input size means the actual resized image size that used for detection, and #proposals means the number of final predicted boxes.

Method	Input Size	Models	RPN Batch Size	Batch Size	#proposals	PRCS (%)	RC (%)	Time (ms)
AdaBoost [14]	380×190	Harr-like	-	-	9025	68.20	41.45	2175
Exhausted CNN [8]	380×190	LeNet (4 layer)	-	256	9025	78.10	56.70	2860
YOLO2 [17]	480×480	Darknet-19 (19 layers)	-	2	196	84.30	77.70	30
Faster-ZF [7]	1000×500	ZF-net (8 layers)	256	256	300	89.50	78.95	37
Faster-VGG [7]	800×400	VGG-16 (16 layers)	256	256	300	90.45	82.65	88
Faster-VGG [7]	1000×500	VGG-16 (16 layers)	256	256	300	91.95	83.15	122
Faster-Res [7]	1000×500	Res-50 (50 layers)	256	256	300	92.35	84.25	334
Proposed-ZF	1000×500	ZF-net (8 layers)	4800	128	60	**93.85**	**90.15**	**29**
Proposed-VGG	1000×500	VGG-16 (16 layers)	4800	128	60	**94.85**	**90.95**	108

stage approaches such as Faster-RCNN can achieve a higher performance in both detection precision and recall than the one-stage YOLO2; 3) our proposed framework can make great improvements to other end-to-end deep learning approaches, especially the recall accuracy. All in all, experimental results have verified the effectiveness and improvements of the proposed method.

There are some other conclusions we can summarize from Table 1. Firstly, benefited from our training framework, our detection task will be significantly boosted under even a simple network instead of adopting complex 16-layer or 50-layer deep networks. From the results of original Faster RCNN, one can realize that using a deeper or better base network can improve both PRCS and RC. However, our proposed method can achieve a much better performance within a 8-layer ZF-net than the original Faster RCNN within a 50-layer Res-net. Secondly, inspired by the intuition that we should learn from normal examples, our method dominates the performance of recall, exceeding others by a large margin. Lastly, our detection system is real-time. For input MMW imaging data, our system takes totally 29 ms for detection, including preprocessing and I/O. Actually, the detection time can be neglected compared to the 2 s computing of our MMW imaging algorithm.

4.3.2. Sensitivities to Hyper-parameters

In Table 2 we investigate the settings of the balanced factor α and the down-weighting parameter γ. We use the default setting of Proposed-ZF in Table 1, then vary α and γ to study sensitivities to our design model and hyper-parameters. Besides, $\alpha = 0.5$, $\gamma = 0$ means the original version of Faster-RCNN (Faster-ZF in Table 1), since if $\alpha = 0.5$ and $\gamma = 0$, L_{rpnfl} will equal L_{rpn}.

In Table 2(a), we set $\gamma = 2$ and vary α from 0.05 to 0.95, and find that $\alpha = 0.25$ performs best. The results are fluctuated marginally (by $\sim 1\%$) when α changes within a wide interval (about [0.25, 0.75]), which demonstrates that α is insensitive for the results in a wide range.

In Table 2(b), α is set to 0.25 by default, since $\alpha = 0.25$ achieves the best results in Table 2(a). Similarly, we add the down-weighting factor γ one by one, from 1 to 5. The results show that $\gamma = 2$ performs best together with $\alpha = 0.25$, and the final performance varies a lot (by $\sim$2–3%) as γ grows.

Table 2. Ablation experiments for different settings of hyper-parameters. All models are trained and tested under the same configurations with Table 1 Proposed-ZF: 1000×500 input size, 8-layer ZF-net, RPNFL, 4,800 RPN batch size. (a) Under different balanced factor α, $\alpha = 0.25$ gain the best performance of both PRCS and RC. (b) Varying the down-weighting factor γ from 1 to 5, $\gamma = 2$ achieves the best accuracy.

(a) Varying α for L_{rpnfl}.

α	γ	PRCS (%)	RC (%)
0.5	0	89.50	78.95
0.05	2	91.79	89.19
0.25	2	**94.82**	**90.09**
0.50	2	**94.82**	89.19
0.75	2	93.00	88.29
0.95	2	92.41	88.19

(b) Varying γ for L_{rpnfl}.

α	γ	PRCS (%)	RC (%)
0.5	0	89.50	78.95
0.25	1	92.61	88.29
0.25	2	**94.82**	**90.09**
0.25	3	93.91	89.29
0.25	4	90.61	88.29
0.25	5	93.35	90.09

4.3.3. Effectiveness of Spatial Map

To validate the effectiveness of our extracted spatial depth map $D(x, y)$, we design several experiments of our models under same settings except the input channels. For comparison, we run two base models ZF and VGG16 on our method, train and test it with different inputs: 1) pure Faster RCNN with one-channel input $I(x, y)$; 2) proposed framework with one-channel input $I(x, y)$; 3) proposed framework with two-channel input $I(x, y)$ and $D(x, y)$.

As shown in Table 3, we design two groups testing with the described 3 input styles, under different base models ZF and VGG16. These results indicate that firstly, within one-channel input $I(x, y)$, the proposed model can make a big progress to the original Faster RCNN; secondly, when utilizing the extracted spatial object map, our detection system can perform much better with an improvement of nearly 3% accuracy in RC, which suggests that it is an effective insight.

Table 3. Comparison of different models with different input channels. Method which has an upper $\star$ denotes it has the same configurations with Proposed-ZF or Proposed-VGG in Table 1. Under a same network, we evaluate and compare different performances w.r.t. our proposed depth spatial map $D(x, y)$.

Method	Input Size	Models	Image Map $I(x, y)$	Spatial Object Map $D(x, y)$	PRCS (%)	RC (%)	Time (ms)
Faster-ZF	1000×500	ZF-net (8 layers)	$\surd$	$\times$	89.50	78.95	37
Proposed-ZF[1]	1000×500	ZF-net (8 layers)	$\surd$	$\times$	93.05	87.35	29
Proposed-ZF*	1000×500	ZF-net (8 layers)	$\surd$	$\surd$	**93.85**	**90.15**	**29**
Faster-VGG	1000×500	VGG-16 (16 layers)	$\surd$	$\times$	91.95	83.15	122
Proposed-VGG[1]	1000×500	VGG-16 (16 layers)	$\surd$	$\times$	93.75	88.25	108
Proposed-VGG*	1000×500	VGG-16 (16 layers)	$\surd$	$\surd$	**94.85**	**90.95**	108

5. CONCLUSION

In this paper, we have presented a novel MMW threats detection system based on MMW imaging techniques and modern deep learning based detection algorithms. Our framework first analyzes the MMW imaging data, then extracts a common MMW image and a new spatial depth map for further detection. This step can obtain the most useful information for next step detection. For detection, we introduce an improved two-stage Faster RCNN method, which is trained through our devised loss function L_{all} for effective learning. Compared with previous works and other state-of-the-art frameworks, our proposed method achieves significant better performance both in precision and recall.

ACKNOWLEDGMENT

This research is partially supported by the National Natural Science Foundation of China under contracts Nos. 61671439, Nos. 61731021, Nos. 81471748.

APPENDIX A. BOUNDING BOX REGRESSION

A.1. Smooth $\mathbf{L_1}$ Function

A general L_1 distance function is a judgement with ℓ_1 norm, denoted as $L_1(\mathbf{t}-\mathbf{t}^*) = \|\mathbf{t}-\mathbf{t}^*\|_1$. However, ℓ_1 function has the undesirable property that its gradient is not continuous, which will disturb the convergence of our network. Instead, we apply a robust smooth L_1 function to compute the loss:

$$\mathrm{F}_{\mathrm{SML1}}(\mathbf{x}) = \begin{cases} 0.5x_i^2 & \text{if } |x_i| < 1 \\ |x_i| - 0.5 & \text{otherwise} \end{cases} \quad \text{for each } x_i \in \mathbf{x}. \tag{A1}$$

And the gradients w.r.t. the input $\mathbf{x}$ is:

$$\mathrm{G}_{\mathrm{SML1}}(x_i) = \frac{\partial \mathrm{F}_{\mathrm{SML1}}(\mathbf{x})}{\partial x_i} = \begin{cases} x_i & \text{if } |x_i| < 1 \\ 1 & \text{if } x_i \geq 1 \\ -1 & \text{otherwise} \end{cases}. \tag{A2}$$

A.2. Parameters Transformation and Regression

For a real bounding box, we use 4 parameters $G = \{x^*, y^*, w^*, h^*\}$ to quantize it. The first two factors represent the left top location of the box, and the last two represent its width and height. The input to our refinement model is a roughly predicted result $P = \{x, y, w, h\}$, and our goal is to learn parameter transformations that map a predicted box P to a ground-truth box G. Following [10], we specify two scale-invariant translations and two log-space translations to refine the box. We parameterize our translation mapping into $\mathbf{t} = \{t_x, t_y, t_w, t_h\}$, where t_x and t_y denote the center translations of scale-invariant, and t_w and t_h denote the width and height translations of log-space, i.e.,

$$\begin{array}{ll} x^* = w \cdot t_x + x & w^* = w \cdot \exp(t_w) \\ y^* = h \cdot t_y + y & h^* = h \cdot \exp(t_h) \end{array}. \tag{A3}$$

Under this definition, the ground-truth targets $\mathbf{t}^*$ for training pair (P, G) are computed as:

$$\begin{array}{ll} t_x^* = (x^* - x)/w & t_w^* = \log(w^*/w) \\ t_y^* = (y^* - y)/h & t_h^* = \log(h^*/h) \end{array}. \tag{A4}$$

REFERENCES

1. Chen, H. M., S. Lee, R. M. Rao, M. A. Slaman, and P. K. Varshney, "Imaging for concealed weapon detection," *IEEE Signal Process. Mag.*, Vol. 22, No. 2, 52–61, 2005.
2. Sheen, D. M., D. L. Mcmakin, and T. E. Hall, "Three-dimensional millimeter-wave imaging for concealed weapon detection," *IEEE Trans. Microw. Theory Techn.*, Vol. 49, No. 9, 1581–1592, 2001.

3. Sheen, D. M., J. L. Fernandes, D. L. Mcmakin, and R. H. Severtsen, "Wide-bandwidth, wide-beamwidth, high-resolution, millimeter-wave imaging for concealed weapon detection," *SPIE Defense, Security, and Sensing*, 871509, 2013.
4. Zhu, Y., M. Yang, L. Wu, Y. Sun, and X. Sun, "Millimeter-Wave holographic imaging algorithm with amplitude corrections," *Progress in Electromagnetics Research M*, Vol. 49, 33–39, 2016.
5. Lowe, D. G., "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, Vol. 60, No. 2, 91–110, 2004.
6. Dalal, N. and B. Triggs, "Histograms of oriented gradients for human detection," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 886–893, 2005.
7. Ren, S., K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 39, No. 6, 1137–1149, 2017.
8. Yao, J., M. Yang, Y. Zhu, L. Wu, and X. Sun, "Using convolutional neural network to localize forbidden object in millimeter-wave image," *J. Infrared Millim. Waves*, Vol. 36, No. 3, 354–360, 2017.
9. Zhu, Y., M. Yang, L. Wu, Y. Sun, and X. Sun, "Practical millimeter-wave holographic imaging system with good robustness," *Chinese Opt. Lett.*, Vol. 14, No. 10, 43–47, 2016.
10. Girshick, R., J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 580–587, 2014.
11. Girshick, R., "Fast R-CNN," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1440–1448, 2015.
12. Lecun, Y., Y. Bengio, and G. Hinton, "Deep learning," *Nature*, Vol. 521, No. 7553, 436–444, 2015.
13. Yeom, S., D. S. Lee, J. Y. Son, and S. H. Kim, "Concealed object detection using passive millimeter wave imaging," *Proc. IEEE Universal Commun. Symp.*, 383–386, 2010.
14. Xiao, Z., X. Lu, J. Yan, L. Wu, and L. Ren, "Automatic detection of concealed pistols using passive millimeter wave imaging," *Proc. IEEE Int. Conf. Imaging Syst. Techn.*, 1–4, 2015.
15. Tapia, S. L., R. Molina, and N. P. Blanca, "Detection and localization of objects in Passive Millimeter Wave images," *Proc. IEEE Signal Process. Conf.*, 2101–2105, 2016.
16. Redmon, J., S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 779–788, 2016.
17. Redmon, J. and A. Farhadi, "YOLO9000: Better, faster, stronger," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 6517–6525, 2017.
18. Krizhevsky, A., I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Proc. Neural Inform. Process. Syst.*, 1097–1105, 2012.
19. Lin, T. Y., P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *Proc. IEEE Int. Conf. Comput. Vis.*, 2999–3007, 2017.
20. Deng, J., W. Dong, R. Socher, and L. J. Li, "ImageNet: A large-scale hierarchical image database," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 248–255, 2009.
21. Rumelhart, D. E., G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," *Readings in Cognitive Science*, Vol. 1, No. 2, 399–421, 1988.
22. Zeiler, M. D. and R. Fergus, "Visualizing and Understanding Convolutional Networks," *Proc. 13th Eur. Conf. Comput. Vis.*, 818–833, 2014.
23. Simonyan, K. and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Proc. Int. Conf. Learn. Representations*, 2015.
24. He, K., X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 770–778, 2016.
25. Jia, Y., E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *Proc. 22nd ACM Int. Conf. Multimedia*, 675–678, 2014.